**Library Management (M.Tech Software Engineering /2022mt93331 )**

| Version Number | Date | Author/Owner | Description of Change |
| --- | --- | --- | --- |
| 01 | 01-August-2023 | SHELKE AKSHAY NANDKUMAR | Problem statement, Requirements Definition, Features identification, Project plan etc. |
| 02 | 02-August-2023 | SHELKE AKSHAY NANDKUMAR | Project features and project plan updated |
| 03 | 20-August-2023 | SHELKE AKSHAY NANDKUMAR | Conceptual Design Phase details updated |
| 04 | 05-Sept-2023 | SHELKE AKSHAY NANDKUMAR | Logical Design Phase details updated |

**I. REQUIREMENT SPECIFICATION**

1. **Problem Statement & Requirements Definition**

***LIBRARY MANAGEMENT SYSTEM:***

The ***Library Management System (LMS)*** aims to streamline and automate the operations of a library, addressing various challenges faced in traditional manual systems. The key problem with manual library management is inefficiency and inaccuracies in tasks such as book borrowing, returning, cataloguing, and tracking book availability. These processes can be time-consuming, error-prone, and lack real-time insights into the library's inventory and usage patterns.

The ***primary goal of the LMS*** is to provide a centralized and user-friendly platform for both librarians and library members to manage library resources efficiently. The system should allow librarians to update, organize, and monitor the library collection, including books, magazines, journals, and other materials. Additionally, library members should be able to search for available resources, place holds, borrow, renew, and return items with ease.

The LMS should include features for generating reports, managing user accounts, tracking overdue items, and handling fine payments. Data security and privacy are also crucial aspects to be considered, ensuring that sensitive information is protected and accessible only to authorized personnel.

To help organize a library, we have decided to create a database. It would be ideal for a library upgrading from a card or paper system where they have a card for each piece of media and for each customer or patron. It would also benefit a library where all employee records are still on a paper basis as well including all hire paperwork and payroll information such as salary. As you can imagine for a paper system, it makes figuring out how many books are overdue, or how many people owe fees, take quite a while to figure out for the employees. Patrons also have no control over the system.

To replace the current card and paper system, we will use this database and it is its main purpose. All tasks previously recorded on paper or cards will be integrated into the new system. For example, based on due dates, librarians can run reports to see who has late books (checked out media report), who owes fees for late books or damaged books (cost report) and much more. It will only take a few seconds to run the report as opposed to going through all the cards by hand, saving the librarians hours a week.

***Roles and Responsibilities of Personas:***

**a) Librarian:** The librarian is responsible for managing the library system, maintaining the database, adding new books, handling user accounts, processing book requests, generating reports, and ensuring the smooth functioning of the LMS.

**b) Library Staff:** The library staff assists the librarian in day-to-day operations, including book shelving, managing returned books, handling book reservations, assisting users with queries, and maintaining a clean and organized library environment.

**c) Library Users (Patrons):** The users have access to the LMS and can search for books, check availability, borrow, renew, and reserve books. They can also pay fines for overdue books and provide feedback on their library experience.

The **Library Management System** is expected to meet following **key business requirements**:

1. **User Authentication:**
   * Secure login and registration for librarians and library users.
   * Role-based access control to restrict actions based on user roles.
2. **Book Management:**
   * Add, update, delete book records with details like title, author, ISBN, & category.
   * Track book availability and location within the library.
3. **Borrowing and Returns:**
   * Allow users to borrow and return books with due date tracking.
   * Automatically calculate and impose fines for late returns.
4. **Book Search:**
   * Provide an advanced search interface with various filters.
5. **Reservation System:**
   * Allow users to reserve books that are currently unavailable.
6. **Reporting and Analytics:**
   * Generate reports on book circulation, popular titles, and user activity.
   * Provide statistics on overdue books and fines collected.
7. **Online Payments:**
   * Enable online fine payment and keep a record of transactions.
8. **Notifications:**
   * Send automated email reminders for overdue books and reservation pickups.
9. **Project features identified:**

|  |  |  |
| --- | --- | --- |
| **Feature ID** | **Feature name** | **Description** |
| **F01** | **User Management** | **This feature allows librarians to manage user accounts, issue library cards, and handle user-related tasks.** |
| **F02** | **Book Cataloging** | **Librarians can add, update, and delete book records with detailed information like title, author, ISBN, etc.** |
| **F03** | **Borrowing and Returns** | **Users can borrow and return books, while the system automatically calculates and imposes fines for late returns.** |
| **F04** | **Online Book Search** | **Users can search for books using various filters like title, author, and category.** |
| **F05** | **Reservation System** | **Users can reserve books that are currently unavailable and get notified when the books become available.** |
| **F06** | **Reporting and Analytics** | **The system generates reports and statistics for administrators to analyze library usage and book circulation.** |
| **F07** | **Online Payments** | **Users can pay fines online securely through the application.** |
| **F08** | **Notifications** | **Automated email reminders are sent to users for overdue books and reservation pickups.** |

1. **Software and hardware details**

|  |  |
| --- | --- |
| **Platform** | **Windows** |
| **Frontend/console** | **JavaScript, Ajax, HTML5, CSS3** |
| **Backend/server** | **PHP (WAMP Server)** |
| **Database** | **MySQL** |
| **Programming Language : Frontend** | **JavaScript, HTML, CSS** |
| **Backend/server: programming Language** | **PHP (WAMP Server)** |

1. **Project Plan:**

The project plan will involve the following stages:

**a) Requirement Specification Document:** Gather detailed requirements from stakeholders and document them.

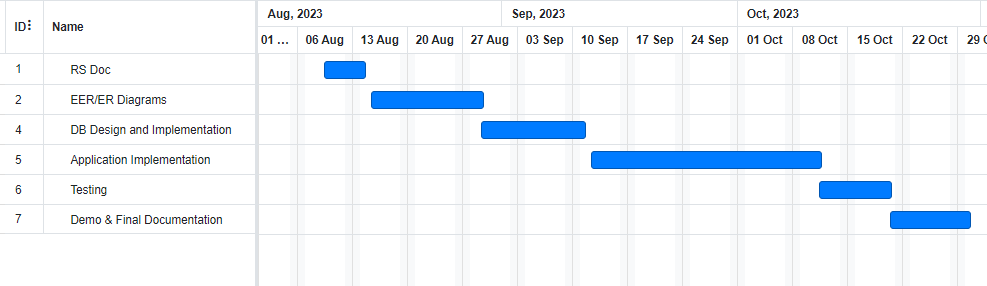
**b) EER/ER diagrams:** Design Entity-relationship diagrams as part of conceptual design phase.

**c) DB Design and Implementation:** Create the database schema, Database Normalization, table/data creation.

**d) Application Implementation:** Implement the application using the chosen technology stack. Design User Interfaces, coding of each feature, system architecture etc.   
Frontend, backend, and database connectivity related implementation.

**e) Testing:** Conduct thorough testing, including unit testing, integration testing, and user acceptance testing, database testing etc.

**f) Demo & Final Documentation:** Gather all project functional statistics and record a demo video. With this information, create and submit final documentation.



1. **Google Drive Link for all deliverables:** [**https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe\_okZsAv?usp=drive\_link**](https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe_okZsAv?usp=drive_link)

**II. CONCEPTUAL DESIGN:**

* 1. A diagram of a student

     Description automatically generated**Entity Relationship Model**
  2. A diagram of a computer program

     Description automatically generated with medium confidence**Object Model Diagram:**
  3. **Google Drive Link for all Deliverables:**[**https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe\_okZsAv?usp=drive\_link**](https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe_okZsAv?usp=drive_link)

**III. LOGICAL DESIGN**

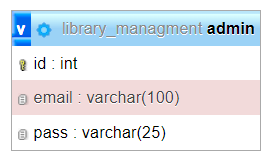
* + 1. A screenshot of a computer

       Description automatically generated**Relational Database Schema**
    2. **Normalization  
       Database Name : library\_management**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Table  Name** | **State** | **Reason** |
| **1** | **issuebook** | **3NF** | **No multivalued attributes, No partial dependencies, No transitive dependency** |
| **2** | **requestbook** | **3NF** | **No multivalued attributes, No partial dependencies, No transitive dependency** |
| **3** | **userdata** | **3NF** | **No multivalued attributes, No partial dependencies, No transitive dependency** |
| **4** | **book** | **3NF** | **No multivalued attributes, No partial dependencies, No transitive dependency** |
| **5** | **admin** | **3NF** | **No multivalued attributes, No partial dependencies, No transitive dependency** |

* + 1. **Create Data Dictionary**

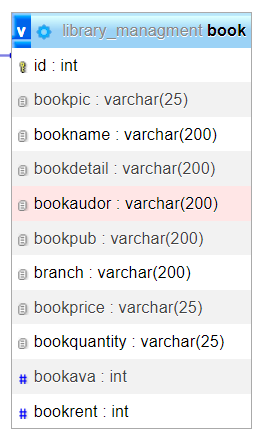
**Table Definitions and Data Contents :**

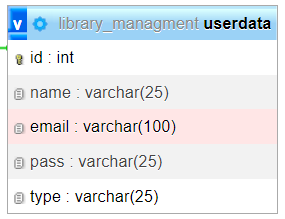
**1. admin table  
**A close up of a logo

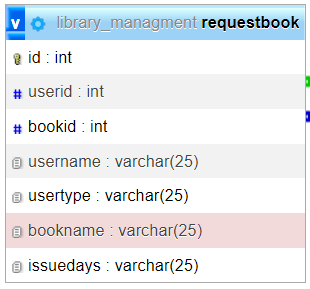
Description automatically generated

**2. book table**

A screenshot of a computer

Description automatically generated****

**3. userdata table  
**A close up of a number

Description automatically generated **4. requestbook table**A screenshot of a computer

Description automatically generated

**5. Issuebook table  
**A screenshot of a computer

Description automatically generated

* 1. **Google Drive Link for all Deliverables:**[**https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe\_okZsAv?usp=drive\_link**](https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe_okZsAv?usp=drive_link)

**IV. PHYSICAL DESIGN**

1. **SQL Statements Database Schema**

|  |
| --- |
| **SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";**  **START TRANSACTION;**  **SET time\_zone = "+00:00";**  **--**  **-- Database: `library\_managment`**  **--**  **--**  **-- Table structure for table `admin`**  **--**  **CREATE TABLE `admin` (**  **`id` int(11) NOT NULL,**  **`email` varchar(500) NOT NULL,**  **`pass` varchar(500) NOT NULL**  **) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**  **--**  **-- Dumping data for table `admin`**  **--**  **INSERT INTO `admin` (`id`, `email`, `pass`) VALUES**  **(1, 'admin@gmail.com', '123');**  **-- --------------------------------------------------------**  **--**  **-- Table structure for table `book`**  **--**  **CREATE TABLE `book` (**  **`id` int(11) NOT NULL,**  **`bookpic` varchar(500) NOT NULL,**  **`bookname` varchar(500) NOT NULL,**  **`bookdetail` varchar(500) NOT NULL,**  **`bookaudor` varchar(500) NOT NULL,**  **`bookpub` varchar(500) NOT NULL,**  **`branch` varchar(500) NOT NULL,**  **`bookprice` varchar(500) NOT NULL,**  **`bookquantity` varchar(500) NOT NULL,**  **`bookava` int(11) NOT NULL,**  **`bookrent` int(11) NOT NULL**  **) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**  **--**  **-- Dumping data for table `book`**  **--**  **INSERT INTO `book` (`id`, `bookpic`, `bookname`, `bookdetail`, `bookaudor`, `bookpub`, `branch`, `bookprice`, `bookquantity`, `bookava`, `bookrent`) VALUES**  **(1, '1.jpg', 'Fundamentals of Database System', '7th edition, 2017', 'Ramez Elamsri and Shamkant Navathe', 'Pearson Education', 'IT', '490', '100', 100, 0),**  **(2, '2.jpg', 'Database System Concepts', '6th edition, 2013', 'Abraham Silberschatz and Henry F Korth and S Sudarshan', 'McGraw Hill', 'IT', '264', '100', 100, 0),**  **(3, '3.jpg', 'An Introduction to Database Systems', '8th edition, 2006', 'Christopher J. Date', 'Addison Wesley', 'IT', '579', '100', 100, 0);**  **-- --------------------------------------------------------**  **--**  **-- Table structure for table `issuebook`**  **--**  **CREATE TABLE `issuebook` (**  **`id` int(11) NOT NULL,**  **`userid` int(11) NOT NULL,**  **`issuename` varchar(500) NOT NULL,**  **`issuebook` varchar(500) NOT NULL,**  **`issuetype` varchar(500) NOT NULL,**  **`issuedays` int(11) NOT NULL,**  **`issuedate` varchar(500) NOT NULL,**  **`issuereturn` varchar(500) NOT NULL,**  **`fine` int(11) NOT NULL**  **) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**  **--**  **-- Dumping data for table `issuebook`**  **--**  **-- --------------------------------------------------------**  **--**  **-- Table structure for table `requestbook`**  **--**  **CREATE TABLE `requestbook` (**  **`id` int(11) NOT NULL,**  **`userid` int(11) NOT NULL,**  **`bookid` int(11) NOT NULL,**  **`username` varchar(500) NOT NULL,**  **`usertype` varchar(500) NOT NULL,**  **`bookname` varchar(500) NOT NULL,**  **`issuedays` varchar(500) NOT NULL**  **) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**  **-- --------------------------------------------------------**  **--**  **-- Table structure for table `userdata`**  **--**  **CREATE TABLE `userdata` (**  **`id` int(11) NOT NULL,**  **`name` varchar(500) NOT NULL,**  **`email` varchar(100) NOT NULL,**  **`pass` varchar(500) NOT NULL,**  **`type` varchar(500) NOT NULL**  **) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;**  **--**  **-- Dumping data for table `userdata`**  **--**  **INSERT INTO `userdata` (`id`, `name`, `email`, `pass`, `type`) VALUES**  **(1, 'akshay\_shelke', '2022mt93331@wilp.bits-pilani.ac.in', '123', 'student'),**  **(2, 'john\_doe', 'johndoe@gmail.com', '123', 'student');**  **--**  **-- Indexes for dumped tables**  **--**  **--**  **-- Indexes for table `admin`**  **--**  **ALTER TABLE `admin`**  **ADD PRIMARY KEY (`id`);**  **--**  **-- Indexes for table `book`**  **--**  **ALTER TABLE `book`**  **ADD PRIMARY KEY (`id`);**  **--**  **-- Indexes for table `issuebook`**  **--**  **ALTER TABLE `issuebook`**  **ADD PRIMARY KEY (`id`),**  **ADD KEY `pk\_fk` (`userid`);**  **--**  **-- Indexes for table `requestbook`**  **--**  **ALTER TABLE `requestbook`**  **ADD PRIMARY KEY (`id`),**  **ADD KEY `pk\_fk\_book` (`bookid`),**  **ADD KEY `pk\_fk\_users` (`userid`);**  **--**  **-- Indexes for table `userdata`**  **--**  **ALTER TABLE `userdata`**  **ADD PRIMARY KEY (`id`);**  **--**  **-- AUTO\_INCREMENT for dumped tables**  **--**  **--**  **-- AUTO\_INCREMENT for table `admin`**  **--**  **ALTER TABLE `admin`**  **MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=2;**  **--**  **-- AUTO\_INCREMENT for table `book`**  **--**  **ALTER TABLE `book`**  **MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;**  **--**  **-- AUTO\_INCREMENT for table `issuebook`**  **--**  **ALTER TABLE `issuebook`**  **MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=12;**  **--**  **-- AUTO\_INCREMENT for table `requestbook`**  **--**  **ALTER TABLE `requestbook`**  **MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=7;**  **--**  **-- AUTO\_INCREMENT for table `userdata`**  **--**  **ALTER TABLE `userdata`**  **MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=6;**  **--**  **-- Constraints for dumped tables**  **--**  **--**  **-- Constraints for table `issuebook`**  **--**  **ALTER TABLE `issuebook`**  **ADD CONSTRAINT `pk\_fk` FOREIGN KEY (`userid`) REFERENCES `userdata` (`id`);**  **--**  **-- Constraints for table `requestbook`**  **--**  **ALTER TABLE `requestbook`**  **ADD CONSTRAINT `pk\_fk\_users` FOREIGN KEY (`userid`) REFERENCES `userdata` (`id`);**  **COMMIT;**  **ALTER TABLE `requestbook`**  **ADD CONSTRAINT `pk\_fk\_book` FOREIGN KEY (`bookid`) REFERENCES `book` (`id`);**  **COMMIT;** |

1. **Stored Procedures/ Triggers/indexes  
   - Please refer above SQL Statements**
2. **User Interface and Database connectivity**

|  |
| --- |
| **Screen #1** |
| **Screen #2** |
| **Screen #3** |
| **Screen #4** |
| **Screen #5** |
| **Screen #6** |
| **Screen #7** |
| **Login Screen** |

|  |
| --- |
| **Database** |
| **Index.php** |
| **DATABASE Connection Code in php** |

1. **Front end or console based codes.**

|  |
| --- |
| **Admin Service Dashboard** |
| **Data class to interact with UI and Database** |
| **Request Book** |
| **Approve Book Request** |

**GIT Hub Link for all code files :**A screenshot of a chat

Description automatically generated

**Link:**

1. **Google Drive Link for all Deliverables:**

[**https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe\_okZsAv?usp=drive\_link**](https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe_okZsAv?usp=drive_link)

**V. FINAL DOC**

1. **TEST CASES for each functionality in the applications.**

**TEST CASES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TASK name/id | testid | TEST DATA | Expected OUTPUT | Actual OUTPUT | TEST PASS/FAIL |
|  |  |  |  |  |  |

1. **DEMO VIDEO for 5 mins.  
   Video Link :** [**www.google.com**](http://www.google.com)
2. **references:**
3. **Google Drive Link for all Deliverables:**

[**https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe\_okZsAv?usp=drive\_link**](https://drive.google.com/drive/folders/14Y7oAMRhSacKphy7nwhFUXvYe_okZsAv?usp=drive_link)